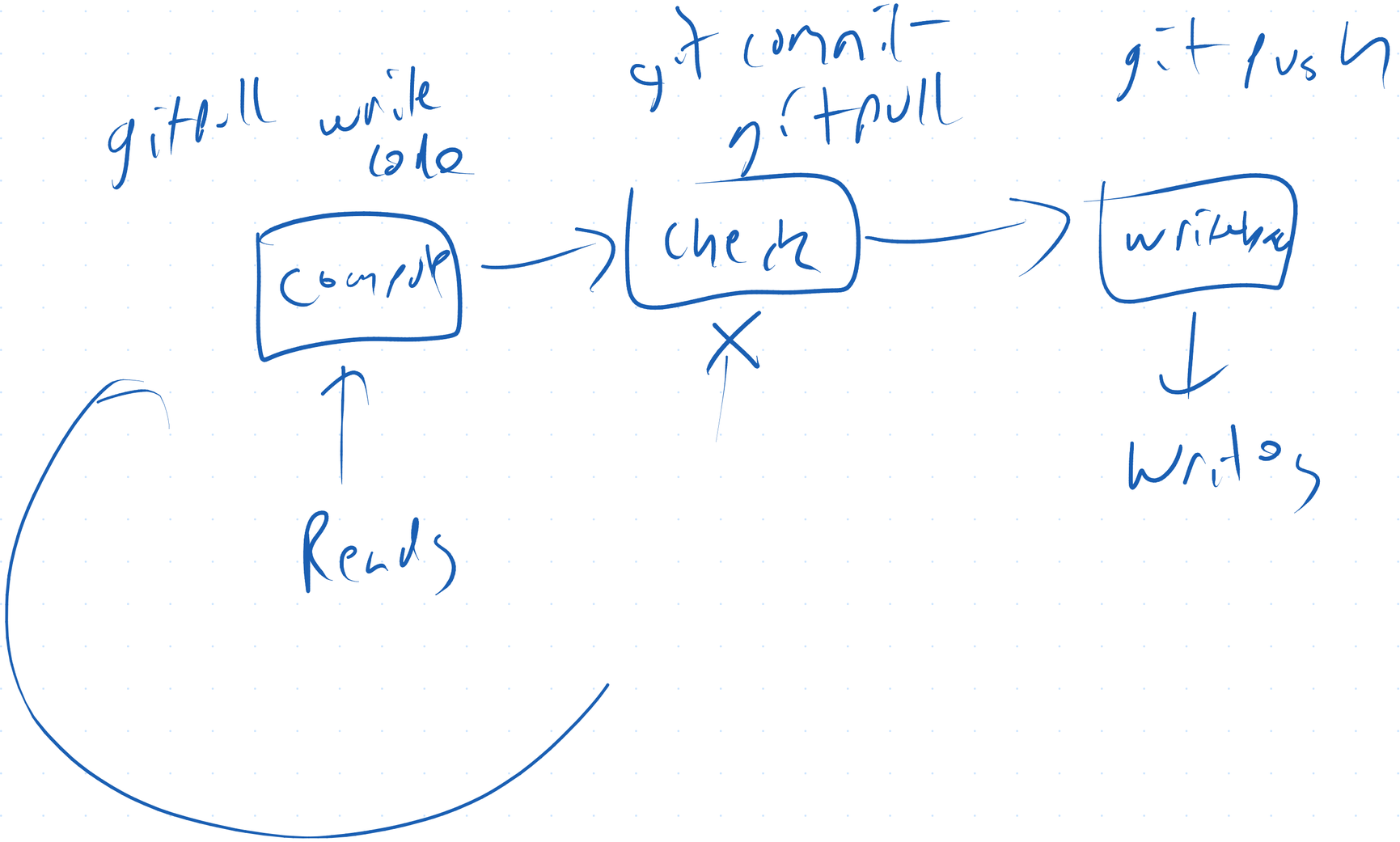
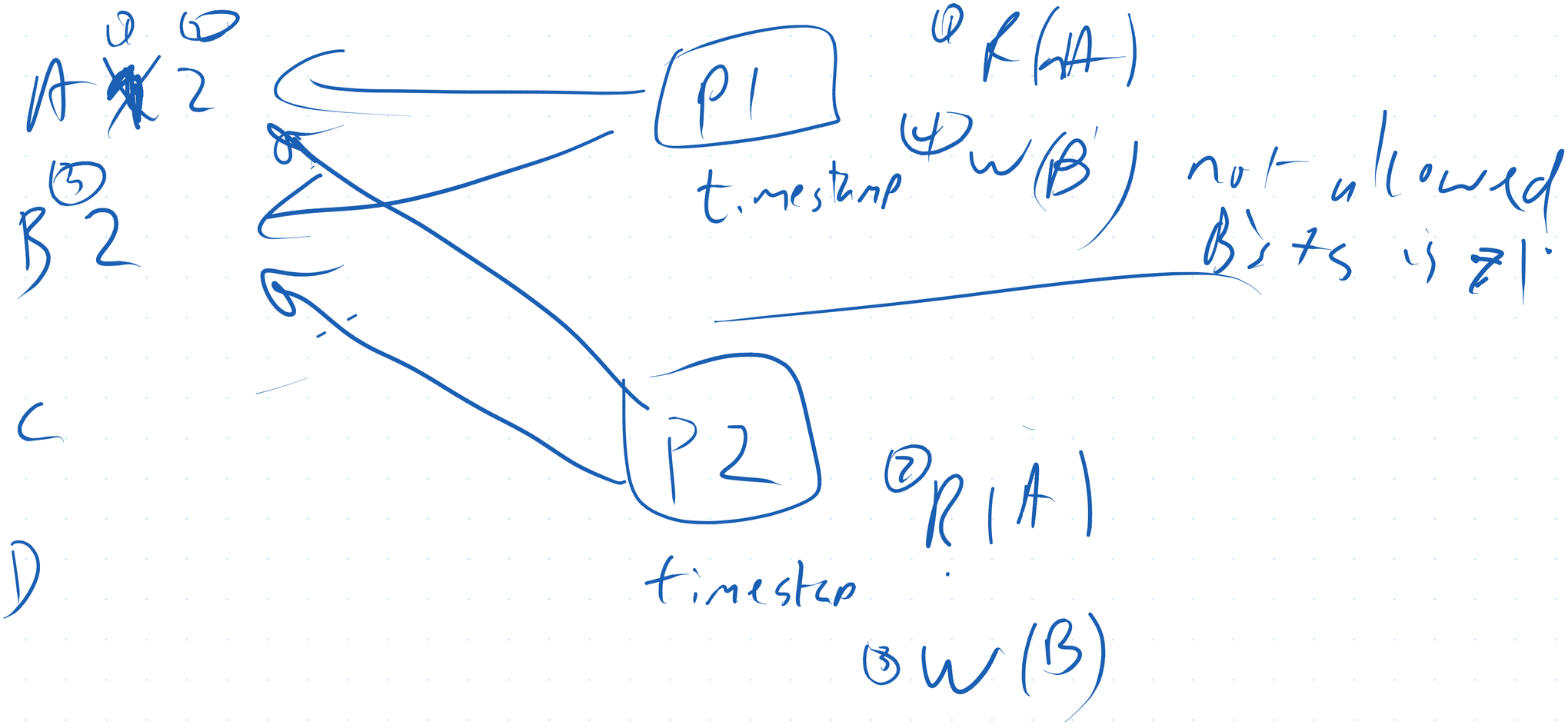

git add write code
git commit-
git pull
git push
Compile
Check
Writing
Writes
Reads




C

D

| | A | B |
|---|---|---|
| 1 | init | init |
| 2 | " | " |
| 3 | " | " |
| 4 | | P2's B |
| | | X |

| | A | B |
|---|---|---|
| P1R(A) | init | init |
| P2R(A) | " | " |
| P1W(B) | " | P1's B |
| P2W(B) | " | P2's B |

## Baseline

Each process has a timestamp ($TS_p$)
Each object has a timestamp ($TS_o$)

on read or write,

if $TS_p \geq TS_o$ → update $TS_o \leftarrow TS_p$
perform op

else abort process

if op is write, just discard the write
Optimization!

TSA TSB

P2 R(A)

P1 R(A)

P1 W(B)

P2 W(B)

1

2

Idea: Dont update TS on read

TSA TSB

P2 R(A) 0

P1 W(A) 1

P1 W(B) 1

P2 W(B) 2

Not ok Write after read error

$RTS_A$ $WTS_A$ $RTS_B$ $WTS_B$

P2 R(A) 2 0

P1 R(A) ✓ 2 0

---

P2 R(A) ② 0

P1 ~~W(A)~~

# Read + Write TS

On Read(O)

↳ If $WTS_O > TS_P$ → Error

Else $RTS_O = Max(RTS_O, TS_P)$

Perform Read

On write(O)

↳ If $RTS_O > TS_P$ → Error

If $WTS_O > TS_P$ → Drop the write

Else $WTS_O = TS_P$

Perform write

P1 W(A)

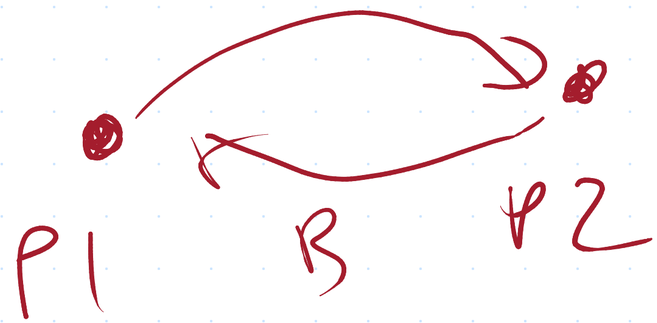
P2 W(A)

P2 W(B)

P1 W(B)

| | $WTS_A$ | $WTS_B$ |
|---|---|---|
| | 1 | 0 |
| | 2 | 0 |
| | 2 | 2 |
| | 2 | 2 |
| | 2 | |

A



## Conflict Equivalence

All conflicts happen in the same order
(No cycles in the happens-before graph)

## View Equivalence

All conflicts happen in the same order excluding writes to the same object
(remove writes to same object from happens before graph)

→ View serializability: View equiv to some serializable schedule